REMARKS

Claims 1-14 are pending in the present application. Claims 2-8 and 13 are amended. Amendments to these claims correct typographical errors that do not change the scope of the claims or add new matter. Support for the claim amendments can be found in the claims as originally filed. Reconsideration of the claims is respectfully requested.

I. Interview Summary

On February 7, 2006, the undersigned attorney and the examiner discussed the anticipation rejection of claim 1 in view of the cited reference. No agreement was reached.

II. 35 U.S.C. § 102, Anticipation

The examiner rejected claims 1, 3, 5, 7, and 14 under 35 U.S.C. § 102(e) as anticipated by *Schuba* et al., Network Protection for Denial of Service Attacks, U.S. Patent 6,725,378 (April 20, 2004) (hereinafter "*Schuba*"). This rejection is respectfully traversed.

Regarding claim1, the examiner states:

> As to independent claim 1, **"A method of preventing a flooding attack on a network server"** is taught in '378 col. 1, lines *55-60* "the present invention includes a unique defense for denial of service attacks";
>
> **"in which a large number of connectionless datagrams are received for queuing to a port on the network server, comprising:"** is shown in '378 cot. 3, lines 16-33 "The Internet Protocol (IP) is the standard network layer protocol of the Internet that provides a connectionless, best effort packet delivery service. IP defines the basic unit of the data transfer used throughout an IP network, called a datagram. The deliver of datagrams is not guaranteed Datagrams are routed towards their destination host" {"connectionless datagrams" same as "connectionless, best effort packet delivery service" / "network server" same as "destination host"}
>
> **"determining, in response to the arrival of a connectionless datagram from a host for a port on the network server"** is disclosed in '378 col. 4, lines 52-54 "When a SYN packet arrives at a port on which a TCP server is listening";
>
> **"if the number of connectionless; datagrams already queued to the port from the host exceeds a prescribed threshold discarding the datagram, if the number of connectionless datagrams already queued**

> **to the port from the host exceeds the prescribed threshold"** is taught in '378 col. 4, lines *54-58* "There is a limit on the number of concurrent TCP connections that can be in a half-open connection state, called the SYN-RECVD state (i.e., SYN received). When the maximum number of half-open connections per port is reached, TCP discards all new incoming connections requests";
>
> **"and queuing the connectionless datagram to a queue slot of the port, lithe number of connection less. datagrams already queued to the port from the host does not exceed the prescribed threshold"** is taught in '378 col. 4, lines *59-67* "until it has either cleared or completed some of the half-open connections".

Office Action of November 14, 2005, pp. 3-4 (emphasis in original).

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). In this case each and every feature of the presently claimed invention is not identically shown in the cited reference, arranged as they are in the claims.

Claim 1 is as follows:

> 1. A method of preventing a flooding attack on a network server in which a large number of connectionless datagrams are received for queuing to a port on the network server, comprising:
>
> determining, in response to the arrival of a connectionless datagram from a host for a port on the network server, if the number of connectionless datagrams already queued to the port from the host exceeds a prescribed threshold;
>
> discarding the datagram, if the number of connectionless datagram already queued to the port from the host exceeds the prescribed threshold; and
>
> queuing the connectionless datagram to a queue slot of the port, if the number of connectionless datagram already queued to the port from the host does not exceed the prescribed threshold.

Schuba does not anticipate claim 1 because *Schuba* does not teach the claimed steps of determining, discarding, and queuing, as claimed. As shown below, the examiner's assertions to the contrary are incorrect.

The examiner asserts that *Schuba* teaches the feature of "determining, in response to the arrival of a connectionless datagram from a host for a port on the network server, if the number of connectionless datagrams already queued to the port from the host exceeds a prescribed threshold," citing from *Schuba* as follows:

> *When a SYN packet arrives at a port on which a TCP server is listening, the above-mentioned data structures are allocated. There is a limit on the number of concurrent TCP connections that can be in a half-open connection state, called the SYN-RECVD state (i.e., SYN received). When the maximum number of half-open connections per port is reached, TCP discards all new incoming connection requests until it has either cleared or completed some of the half-open connections.* Typically, several ports can be flooded in this manner, resulting in degraded service or worse. Moreover, it should be appreciated that without a limit on the number of half-open connections, a different denial of service attack would result in which an attacker could request so many connections that the target machine's memory is completely exhausted by allocating data structures for half-open TCP connections. Table II illustrates the half-open connection states that may be accommodated by various operating systems as follows:
>
> TABLE II
>
> | Operating System | Backlog | Backlog + Grace |
> |---|---|---|
> | FreeBSD 2.1.5 | n.a. | 128 |
> | Linux 1.2.x | 10 | 10 |
> | Solaris 2.4 | 5 | n.a |
> | Solaris 2.5.1 | 32 | n.a. |
> | SunOS 4.x | 5 | 8 |
> | Windows NTs 3.51 | 6 | 6 |
> | Windows NTw 4.0 | 6 | 6 |

Schuba, col. 4, l. 52 through col. 5, l. 13 (emphasis to show portions cited by the examiner).

The examiner uses the same portion of *Schuba* to support the examiner's other assertions. However, the cited portion of *Schuba* teaches a method of defeating flooding attacks by limiting the number of half-open connections allowed at a given port. *Schuba* defines a half-open connection as a state in which the SYN datagram from a destination host has been received at a source host. As shown below, a half-open connection is not the same as a queue of datagrams, contrary to any assumptions or assertions the examiner has made. Thus, *Schuba* does not teach "determining, in response to the arrival of a connectionless datagram from a host for a port on the

network server, *if the number of connectionless datagrams* already queued to the port from the host *exceeds* a prescribed threshold," as claimed in claim 1. Accordingly, *Schuba* does not anticipate claim 1.

As shown below, a half-open connection is not the same as a queue of datagrams. *Schuba* describes the process of establishing a transmission control protocol (TCP) connection in figure 1 of *Schuba*. Specifically, a three-way handshake occurs between a source host and a destination host. *Schuba* specifically states that before data can be transmitted between a source host and a destination host, a connection must be established between the two hosts. The source host sends a SYN datagram to the destination host. The destination host then sends a SYN+ACK datagram to the source host. At this point, the destination host is in the SYN_RECVD state. In this state, the destination host is waiting for an ACK datagram from the source host. The source host then sends an ACK datagram to the destination host, resulting in a connected state between the destination host and the source host. Thereafter, additional datagrams containing data are exchanged between the source host and the destination host. *Schuba* describes this process generally as follows:

> Referring to FIG. 1, a diagram is provided that illustrates the TCP packet sequence of a three-way handshake needed to establish a TCP connection. Before data can be transmitted between a source host S and a destination host D, TCP needs to establish a connection between source host S and destination host D. The connection establishment process is called the three-way handshake. The three-way handshake is established by exchanging certain TCP packet types between source host S and destination host D. The TCP packet types are distinguished by dedicated flag bits set in a TCP header code field and are listed in Table I as follows:
>
> TABLE I
> TCP Header Flag Bits
>
> | | Abbreviation |
> |---|---|
> | SYNchronize | SYN |
> | ACKnowledgement | ACK |
> | ReSeT | RST |
>
> It should be appreciated that, under appropriate conditions, more than one of the flag bits may be set in the same TCP packet.
>
> The first transmission in the three-way handshake is from source host S to destination host D in the form of a SYN packet (SYN flag bit set) while destination host D is in the LISTEN state. The second message, from destination host D to source host S, has both the SYN and ACK bit flags

> set (SYN+ACK) indicating that destination host D acknowledges the SYN packet and is continuing the handshake. At this point, destination host D is in the SYN_RECVD state. The third message, from source host S to destination host D has its ACK bit flag set, and is an indication to destination host D that both hosts S and D agree that a connection has been established, resulting in the CONNECTED state of destination host D. The third message may contain user payload data. Datagrams D1 and D2 represent data exchanges that take place after proper establishment-of the TCP connection.

Schuba, col. 3, l. 46 through col. 4, l. 16.

As stated above, *Schuba* defines a half-open connection as a state in which the SYN datagram from a destination host has been received at a source host. Thus, the destination host is waiting for an ACK datagram from the source host. If the source host never sends an ACK datagram, then no full connection is established. However, the destination host still holds a half-open connection at a port in anticipation of receiving the ACK message. If the ACK message never arrives, then the port at the destination host can become unavailable. A malicious user can take advantage of this fact by causing the destination host to form enough half-open connections that no ports remain available to genuine users. *Schuba* specifically deals with this problem by limiting the number of half-open connections that are available at a given port at a destination host.

In contrast, the invention of claim 1 limits the *number of datagrams* that are allowed to *queue* at a given port. Specifically, claim 1 provides that an arriving datagram should be discarded "if the number of connectionless datagram already queued to the port from the host exceeds the prescribed threshold." *Schuba* does not teach this claimed feature. *Schuba* teaches limiting the number of half-open connections by discarding new requests for connections. *Schuba* does not teach discarding datagrams to be assigned to a queue.

Similarly, *Schuba* does not teach "determining, in response to the arrival of a connectionless datagram from a host for a port on the network server, *if the number of connectionless datagrams already queued* to the port from the host exceeds a prescribed threshold." Instead, *Schuba* counts the number of *half-open connections* at a port, which as described above, is entirely different than determining the number of connectionless datagrams already queued to a port.

Similarly, *Schuba* does not teach "queuing the connectionless datagram to a queue slot of the port, if the number of connectionless datagram already queued to the port from the host does not exceed the prescribed threshold." *Schuba* teaches establishing connections and limiting the number of half-open connections, not queuing datagrams at a port if the number of connectionless datagrams already queued to a port has not been exceeded.

In summary, *Schuba* does not teach the features of claim 1 because *Schuba* does not teach anything regarding determining the number of connectionless datagrams queued at a port or discarding or queuing the number of connectionless datagrams at a port. Instead, *Schuba* teaches determining the number of half-open connections at a port. The two features are entirely distinct for the reasons given above. Accordingly, *Schuba* does not anticipate claim 1.

Claims 3, 5, 7, and 14 all contain features similar to those presented in claim 1. Hence, *Schuba* does not anticipate these claims for the same reasons presented above. Therefore, the rejection of claims 1, 3, 5, 7, and 14 under 35 U.S.C. § 102 has been overcome.

Furthermore, *Schuba* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. Absent the examiner pointing out some teaching or incentive to implement *Schuba* and limiting the number of connectionless datagrams queued at a port, one of ordinary skill in the art would not be led to modify *Schuba* to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify *Schuba* in this manner, the presently claimed invention can be reached only through an improper use of hindsight using Applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

III. 35 U.S.C. § 103, Obviousness

The examiner rejected claims 2, 4, 6, and 8-13 under 35 U.S.C. § 103(a) as obvious over *Yavatkar* et al., Method and System for Diagnosing Network Intrusion, U.S. Patent 6,735,702 (May 11, 2004) (hereinafter "*Yavatkar*"). This rejection is respectfully traversed.

Regarding claim 2, the examiner states that:

> As to dependent claim 2, the following is not taught in '378 "wherein the determining if the number of datagrams already queued to the port from the host exceeds a prescribed threshold further comprises: calculating the prescribed threshold by multiplying a percentage P by the number of available queue slots for the port" however '702 teaches "A watchdog agent may assume a network attack exist if network congestion is detected

> ... In an alternate embodiment a watchdog agent detects network congestion by monitoring interface discard counts and average queue lengths for each port on the node" in col. 15, line 63 through col. 16, line 17.
>
> It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of '378 a method to protect a network from denial of service attacks to include a means to calculate the threshold limit per port. One of ordinary skill in the art would have been motivated to perform such a modification in order to gain information needed to diagnose a network attack (see '702 col. 2 lines 44 et seq.) "Therefore there exists a need for a system and method allowing for the distributed state of a network such as information about attack traffic, to be quickly and accurately collected. A system and method are needed for quickly and accurately diagnosing network attacks by determining information such as the source of, or a partial path of, attack traffic".

Office Action of November 14, 2005, pp. 5-6.

The examiner asserts that *Yavatkar* teaches the remaining features of claims 4, 6, and 8-13, but relies on the same motivation to combine *Yavatkar* and *Schuba* as asserted for claim 2.

If the Patent Office does not produce a *prima facie* case of unpatentability, then without more the applicant is entitled to grant of a patent. *In re Oetiker*, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992); *In re Grabiak*, 769 F.2d 729, 733, 226 U.S.P.Q. 870, 873 (Fed. Cir. 1985). A prima facie case of obviousness is established when the teachings of the prior art itself suggest the claimed subject matter to a person of ordinary skill in the art. *In re Bell*, 991 F.2d 781, 783, 26 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1993). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). In this case, the examiner has not established that all features of the claimed invention have been considered.

The examiner has failed to state a *prima facie* obviousness rejection because the proposed combination does not teach all of the features of the claims. Claims 4, 6, and 8-13 depend from independent claims 1, 3, 5, or 7. As shown above, *Schuba* does not teach all of the features of the independent claims. Furthermore, *Schuba* does not suggest the features of the independent claims because *Schuba* describes an entirely different method of dealing with flooding attacks compared with the method of claim 1 and with the features of the other independent claims. The examiner tacitly admits that *Yavatkar* does not teach the features of the independent claims because the examiner does not assert otherwise and because the examiner would not otherwise

need to rely on *Schuba*. Thus, the proposed combination does not teach or suggest all of the features of the independent claims. Accordingly, the examiner has failed to state a *prima facie* obviousness rejection against claims 4, 6, and 8-13 at least by virtue of their dependence on the respective independent claims.

In addition, the proposed combination does not teach all of the features of the other dependent claims. For example, neither *Yavatkar* nor *Schuba* teach or suggest the feature of "configuring a maximum number of connectionless datagrams allowed to be queued at the port," as claimed in claim 9. The examiner tacitly admits that *Schuba* does not teach this claimed feature. The examiner states that *Yavatkar* does teach this claimed feature as follows:

> **As to dependent claim 9, "further comprising: configuring a maximum number of connectionless, datagrams allowed to be queued at the port"** is taught in '702 col, 12, lines 27-39 "In step 440, proactive environment 100 instantiates service object 300 based on the class of service 102. Proactive environment 100 configures service object 300 per the permissioning accessed in step 434. For example, one set of permissioning may allow agent 110 to use service object 300 to read the operating characteristics of port 21 and alter settings for the port".

Office Action of November 14, 2005, p. 6 (emphasis in original).

The portion of *Yavatkar* cited by the examiner is as follows:

> In step 440, proactive environment 100 instantiates service object 300 based on the class of service 102. Proactive environment 100 configures service object 300 per the permissioning accessed in step 434. For example, one set of permissioning may allow agent 110 to use service object 300 to read the operating characteristics of port 21 and alter settings for the port, and another set of permissioning may allow agent 110 to use service object 300 only to read the operating characteristics of port 21. Proactive environment 100 sets permission variables 312, members of service object 300, to indicate which aspects of service 102 (in the form of methods 322-326 of service object 300) agent 110 may access.

Yavatkar, col. 12, ll. 27-39.

The cited text plainly does not teach or suggest "configuring a maximum number of connectionless datagrams allowed to be queued at the port," as claimed in claim 9. Nothing else in *Yavatkar* teaches or suggests this claimed feature. Because neither *Schuba* nor *Yavatkar* teach or suggest this claimed feature, the examiner has failed to state a *prima facie* obviousness rejection of claim 9.

In summary, the examiner has failed to state a *prima facie* obviousness rejection against any of the claims because the proposed combination does not teach or suggest all of the features of the claims. Accordingly, the rejection of claims 2, 4, 6, and 8-13 has been overcome.

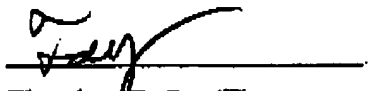
IV. Conclusion

It is respectfully urged that the subject application is patentable over *Schuba* and is now in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: February 14, 2006

Respectfully submitted,



Theodore D. Fay III
Reg. No. 48,504
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants